

---

# Data Processing Subsystem PRONG CSC

**Jianhe Zhang**

[jzhang@eos.hitc.com](mailto:jzhang@eos.hitc.com)

---

**16 April 1996**

# PDPS Roadmap



*Special Topic: Production Rules*

Capture PGE Profile at **SSI&T**

Describe Production Goals through **Production Requests**

Accept **On-demand** Production Requests

Accept Resource Reservations and Create **Resource Plans**

Planning Production Controls - Create and Activate **Production Plans**

Coordinate Production from Data Arrival with **Subscription** Notifications

Handle L0 **Data Preparation**

*Special Topic: Production Subsetting*

Realtime **Production** Controls and PGE Execution Monitoring

*Special Topic: PGE Exit Handling*

**Quality Assurance** Check Output Products

*Special Topic: PDPS Database*

*Special Topic: Ancillary Data Pre-Processing*

# Design Drivers



## General Functional Description

- Job Creation, Submission and Control
- Resource Allocation, Deallocation and Monitoring
- Data Staging and Destaging

# Design Drivers (cont.)

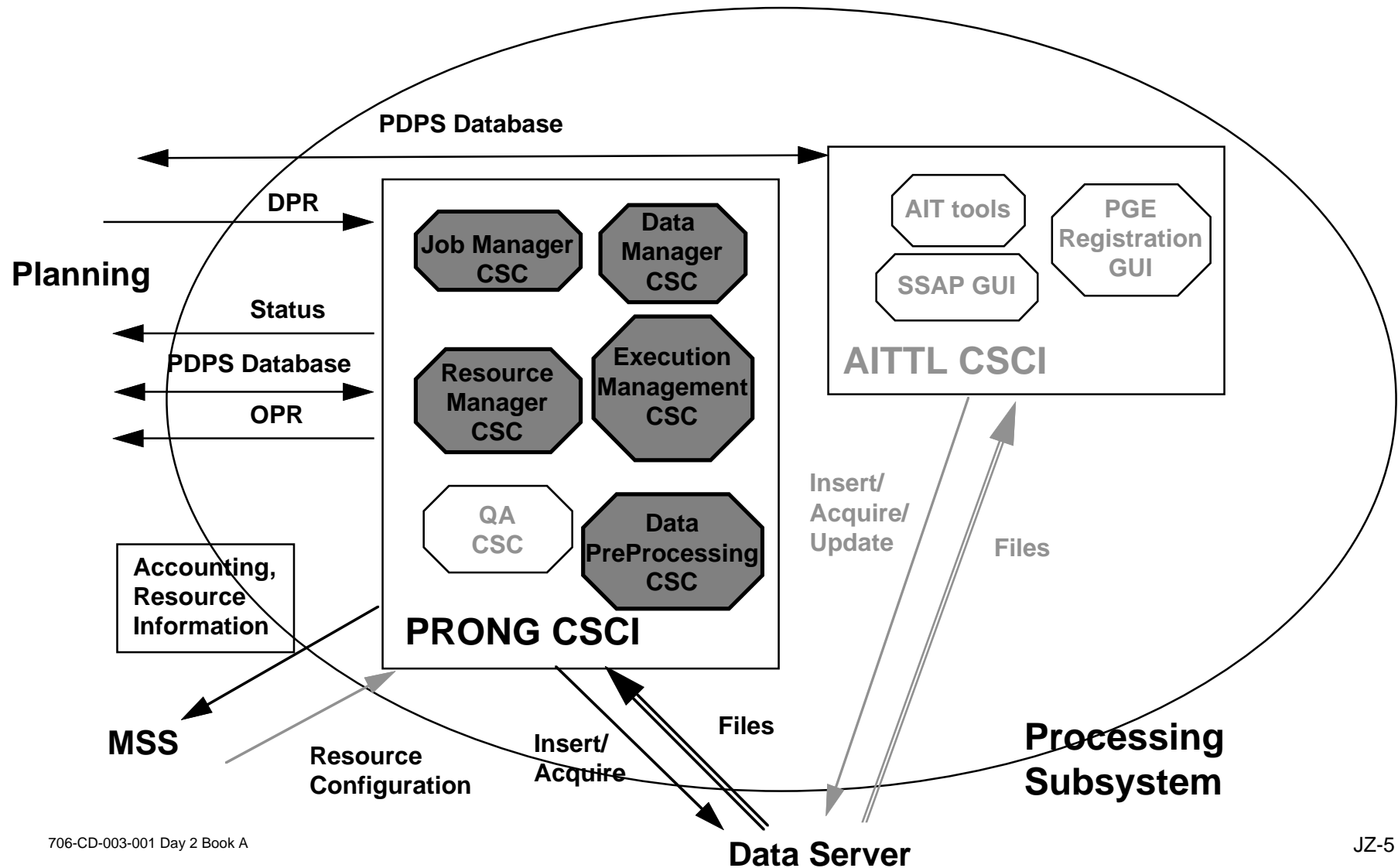


## New Release B Features

- **Predictive Staging**
  - Using predictions to start some data staging ahead of time so that some input data is already on the local disk when a DPR is ready to be released
- **Resource Queue Management**
  - Queuing of jobs waiting for resources
- **Late Start Time Monitoring**
- **Job Load Management**
  - Handling the large number of DPRs under Release B baseline
- **Error Handling (Special Topic not included here)**

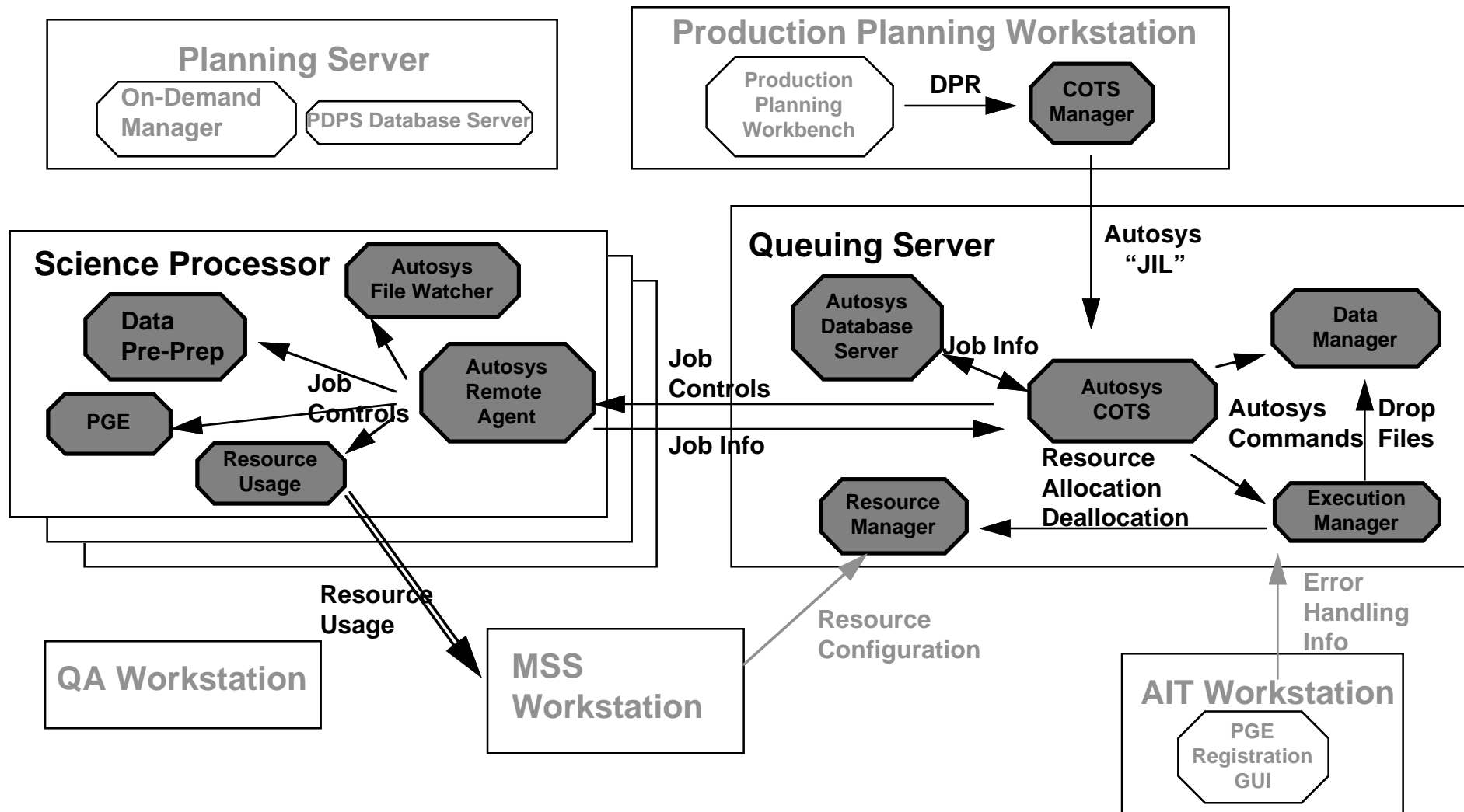


# HW & SW Architecture





# HW/SW Architecture



# Public Interfaces/Key Mechanisms



## Data Server Interface

- **Acquire PGE and Input Data Granules From Data Server**
- **Insert Output Data Granules To Data Server**

## Key Mechanisms

- **Universal References**
- **Process Framework**

# Predictive Staging



## Object Model:

- **Job Management**

305-CD-027-002 4.3.2

## Event Traces:

- **Create a Data Processing Request Job** 305-CD-027-002 4.5.2.1
  - This scenario describes how the predictive staging job and the job box of a DPR are created when planning subsystem calls data processing subsystem to create a data processing request job.
- **Predictive Staging / Make Data Local Job** 305-CD-027-002 4.5.3.8
  - This scenario describes how some of the input data of a DPR is predictively staged from data server to the local disk before a PGE is released.





# Predictive Staging (cont.)

**PDL:**

- **EcUtStatus DpPrScheduler::CreateDprJob(Dpr : PIDPR&)**  
305-CD-027-002
- This method is used to convert data processing request to Autosys managed jobs. If the predictive staging flag is set, the predictive staging job is created here.
- **void DpPrDataManager::MakeDataLocal(DPRid : int, Machine : RWCString)**  
305-CD-027-002
- This method predictively stages data from the data server to the local disk and records data staging statistics



# Resource Queue Management

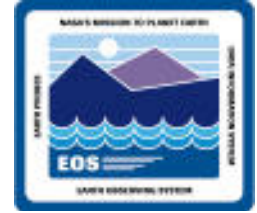
## Object Model:

- **Execution Management**

## Event Traces:

- **DeQueue a Job Awaiting Resources**
  - This scenario describes how a job waiting for resources is selected to be taken “off hold” when the needed resources are deallocated.

# Resource Queue Management (cont.)



**PDL:**

- **Void DpPrWaitingResourceJobNB::DeQueueWaitingJob(machine : RWCString)**
  - This method tries to dequeue a job waiting for resources. If the resources are in a competing state and no jobs can be dequeued, resource allocation is performed by deallocating selected candidates. If low priority jobs can be dequeued while high priority jobs have waited “too long,” because the resources are not available, the low priority jobs will not be dequeued and the available resources will be set aside for the high priority jobs.

# Late Start Time Monitoring



## Object Model:

- **Job Management**

## Event Traces:

- **Monitor a Late Start Job**
  - This scenario describes how the data processing subsystem monitors the data processing jobs which are delayed for executions.

## PDL:

- **void StartTimeCheckerNB::AlarmDelayedDPRs(void)**
  - This method checks all jobs in the database and alarms the operator if any of them have waited “too long.”



# Job Load Management

## Background

- Release B processing design needs to handle the job load specified in Release B baseline.

## Approaches

- DPS design adopted the following approaches to deal with this issue
  - Delayed job creation:  
The PGE preparation, PGE and post PGE jobs are not created until after resources to run the PGE are available.
  - Multiple Autosys instances:  
Install multiple Autosys instances to manage one or more processing clusters. The CotsManager dispatches jobs to the right Autosys instance according to the target machine of the jobs.
  - Control Job Loading:  
Load jobs from PDPS database if the number of jobs in Autosys falls below certain threshold.

# Job Load Management (cont.)



## Object Model:

- **Job Management**

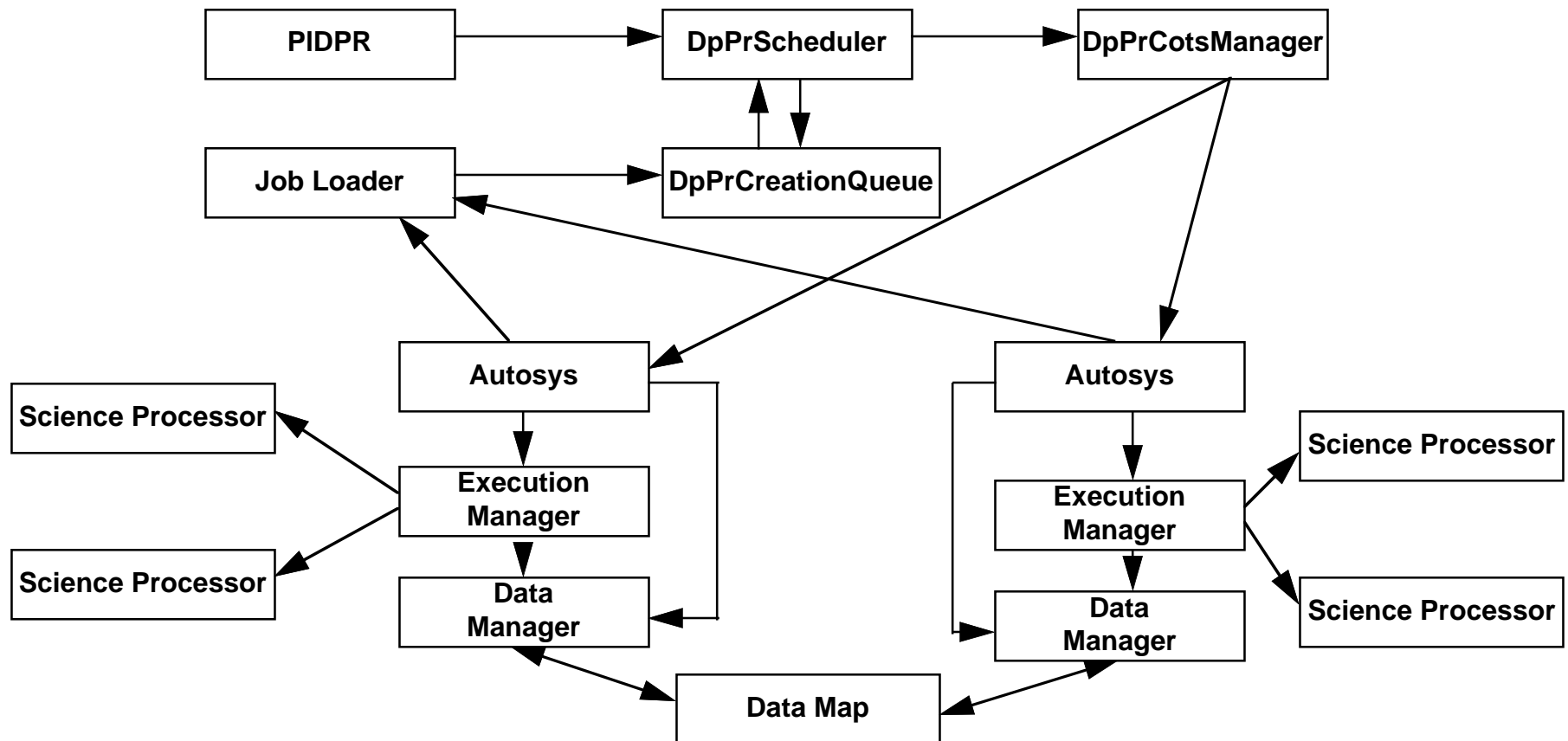
## Event Traces:

- **Expand Job Box**
  - This scenario describes how the job box are expanded after the resource allocation job is completed successfully. Note: if resource allocation job fails, it puts itself in to resource waiting queue.
- **Control Job Loading**
  - This scenario describes how a DPR is queued for creation and how the job loader is started by Autosys to load the DPR in to Autosys database.



# Job Load Management (cont.)

## Multiple Autosys Instance Model



# Summary



- **Predictive Staging** provides a way to stage input data before a PGE is released. It adjusts its prediction accuracy of the start time for the predictive staging job using historical data stored after each time data is predictively staged.
- **Modified resource management** handles resources efficiently and provides user configuration features to resolve resource competing situations.
- **Late start time Monitoring** provides an extra feature for production operator to be notified of late start jobs.
- **Load Management** adds expandable and configurable features for DPS to handle job load.